Desalination and Water Treatment
www.deswater.com

doi: 10.1080/19443994.2013.792009

52 (2014) 1272–1286
February

Taylor & Francis
Taylor & Francis Group

A UPCA-based monitoring and fault detection approach for reverse osmosis desalination plants

D. Garcia-Alvarez*, M.J. Fuente

Department of Systems Engineering and Automatic Control, School of Industrial Engineering, University of Valladolid, C/Real de Burgos S/N 47011 Valladolid, Spain
Tel. +34 983 42 35 63; Fax: +34 983 423161; email: dieggar@cta.uva.es

Received 10 December 2012; Accepted 21 March 2013

ABSTRACT

This article studies and describes a monitoring, fault detection, and diagnosis technique based on the unfolded PCA (UPCA) approach and its application to a reverse osmosis desalination plant. The UPCA approach is normally applied to batch processes, but in this case, the UPCA approach is applied to a continuous process, which does not present a strict steady state. The classical principal component analysis (PCA) approach is not very suitable for this process due to the nonlinearities of this type of processes. The principal characteristics of PCA and UPCA methods are described. The different considerations and adaptations required to perform a UPCA monitoring tool applied to a continuous process, such as unfolding, alignment, and imputation, are also described and explained.

Keywords: PCA; UPCA; Fault detection and isolation; Desalination plant

1. Introduction

One of the most important objectives of modern industry is production with the best quality. This aim can be compromised by the appearance of special causes in the process. These special causes can sometimes be due to faults in the process. The appearance of these faults can put the health of the plant operators, the final users, or the environment at risk. The modern control theory has solved several problems with optimal results from the point of view of quality. However, automatic monitoring and fault detection schemes have to be designed and implemented to detect and diagnose the faults.

A fault can be defined as a deviation of the nominal situation in the structure or parameters of the system [1]. A review of the principal fault detection and isolation (FDI) techniques is given in [2–4]. The authors divide this area into three categories: quantitative model-based methods, qualitative model-based techniques, and process history-based methods. The authors describe the main advantages and drawbacks of using each of these techniques. In this work, the technique applied to perform a monitoring tool is a statistical quantitative process history-based technique; specifically, the principal component analysis (PCA) is the chosen approach.

The PCA is based on a linear transformation that calculates new uncorrelated variables (components) from the correlated original measured variables. A few of these components are enough to represent the sources of variability in the process. This characteristic makes PCA a suitable tool for system monitoring [5].

*Corresponding author.

Also, the PCA can detect process variations by using monitoring statistics, which have been used as fault detection schemes [6].

The PCA approach has been widely used in monitoring and fault detention tasks in continuous processes [5,6]. The PCA monitoring tools obtain good results when used to monitor steady states, since the relationship between the variables in the steady states are linear. However, the classical PCA fault detection schemes are not very suitable for processes that present nonlinear behavior, because an increment in the false alarms ratio can be observed, due to a change in the correlation structure due to the nonlinear relationship between the variables.

In order to deal with these nonlinear situations, several variations of the classical PCA are described in the literature, for example, the adaptive PCA approach [7], the recursive PCA approach [8], the exponentially weighted PCA approach [9], or the nonlinear PCA approach based on auto-associative artificial neural networks [10].

None of these modifications are suitable when transient states in the process are long and persistent, i.e. the desalination plant explained in this work. This process, despite of being a continuous process, does not present a constant steady state due to the cleaning cycles.

There is a configuration of the classical PCA approach to monitor normally nonlinear processes: the batch processes. These methods are called multi-way PCA, unfolded PCA (UPCA), or Batch PCA [11–14]. All of them are equivalent.

This article describes a FDI method based on the UPCA scheme. The designed approach is applied, as cited before, to a simulated reverse osmosis (RO) desalination plant. This model is based on small and medium real plants placed in remote areas. This remote location requires the use of monitoring tools, since the operator cannot be in the operation room all day. The use of modern technologies can allow the state of the plant to be monitored from a remote centralized operation center.

Recently, several works related to the fault detection in RO desalination plans have been published. In Ref. [15], a model-based approach is presented. The authors established a nonlinear mathematical model. The model output is compared with the real output in order to detect faults in the process. The execution of the complex model in required at every sample time in the online running. In Ref. [16], the use of models is also proposed, but, in this case, the model is based on bond graphs. This configuration can be too complex for real-plant implementations. In Refs. [17] and [18], the fault tolerant control task is undertaken. In these works, the authors do not implement any fault detection method, but this configuration requires a fault detection method. Bourouni [19] proposed to analyze the plant availability using graphical methods such as the reliability block diagram method and the fault tree analysis method.

PCA and UPCA approaches can be applied to several processes without implementing complex mathematical model, based only in data collected from the plant. The study case of this work is a continuous process, but due to the cleaning phases for the correct plant running, its behavior is not precisely a steady state. So, the designed monitoring tool is based on the UPCA scheme.

This paper tests the adaptation of the UPCA approach to a continuous process in order to reduce the false alarms ratio and to improve the detectability. This work includes the necessary considerations and steps for designing and implementing a monitoring tool based on UPCA applied to continuous processes without a strictly steady behavior. The paper describes the principal tasks to be performed by the UPCA, i.e. the unfolding, the alignment, and the imputation.

The monitoring and fault detection of nonsteady behaviors have been studied by several authors. In Refs. [20] and [21] several approaches based on UPCA are presented and discussed. However, they do not deal with the imputation problem, i.e. the missing data that appear when this methodology is applied in online process monitoring, or they use the trimmed scores (TRI) imputation method. In this work, other imputation method is used with better results and a similar computational cost. Also, an online alignment data problem, necessary in UPCA approaches, is implemented and used following the guidelines presented in [22]. In the identification of the variables related to the fault task, a contribution plot organization for the variable contributions to the scores are proposed by grouping the scores grouped by each process variable.

The study case of a desalination plant is presented in section 2. The PCA as an FDI technique is presented in section 3. Section 4 describes the UPCA approach and the main variations that must be considered when designing a FDI tool for a continuous process. The conclusions of this work are presented in section 6.

The mathematical notation used in this paper is the following: bold capital letters for matrices, bold lower case letters for vectors, and italics letters for scalars.

2. The RO desalination plant

The approach of FDI presented in this work has been applied to a RO desalination plant. This plant is widely explained in [23]. The plant has been developed as one of the results of the European project OPEN-GAIN (http://www.open-gain.org/).

The proposed approach has been tested in a simulation model of this plant. The plant has been simulated using the simulation environment EcosimPro©. EcosimPro© is a dynamic simulation tool based on an object-oriented modeling language. This type of tools allows first principles models to be built. Every real component in the system can be modeled as a logical component. The physical parameters, such as the quality of feed water, salinity, temperature, type of filter, membrane characteristics, etc. can be included in the model. The simulation model of the RO desalination plant consists approximately of 9,000 equations and 9,000 variables and it has been validated using real data.

The principal aim of the plant used in this work is to desalinate brackish water. The brackish water is pumped from the well to a supply tank. The water of the supply tank is pumped through a high pressure pump. The objective of this pump is to increase the pressure to above the osmotic pressure. The pressurized water goes through the RO membrane rack. This difference in pressure between the membrane's sides creates a flow of clean water. The clean water is stored in another tank after a purification process. This last tank supplies water to the consumers.

The simulated plant used in this work corresponds to a real plant placed in a remote area in Tunisia, which is why a simulated plant is used. The plant is a prototype and is not running normally yet and, so there are not enough data to perform this type of data-driven FDI tasks. The aim of the simulated plant is to test different techniques as control or monitoring and fault detection methods, in order to facilitate its autonomous functioning and to reduce the human maintenance and operation due to its location. An overview of this plant can be seen in Fig. 1.

Another reason to use a simulated plant is because several types of faults can be included by manipulating the equations or parameters of the different physical components.

For example, different types of breakage can be simulated in the membrane or different types of blockage in the different filters without taking any risk. The plant is based on a RO separation process. It is necessary to use high pressure to force the water through a semi-permeate membrane. The membrane retains the salt. Two different filters are placed before the membrane: first a sand filter and then a cartridge filter. These filters are required to remove several types of solid particles which can damage the membrane.

The decrease in the performance of membranes and filters during the plant operation is common in this type of plants. This is due to several types of deposits, such as scale, organic components, silt, etc. Cleaning cycles are run to clean these deposits in order to obtain an optimal plant operation and avoid possible malfunctions.

The accumulation of deposits in the different filters and membranes and the required cleaning cycles are the reasons why the plant does not strictly run in a steady state. This is due to the noticeable differences in several of the pressure and concentration measurements when the plant has just been cleaned and when the plant has been cleaned a long time ago. Fig. 2 shows this phenomenon in one of the pressures measured in the membrane input.

Several variables were considered in this work. All variables correspond to variables measured in the real plant in control loops or for supervision tasks. The

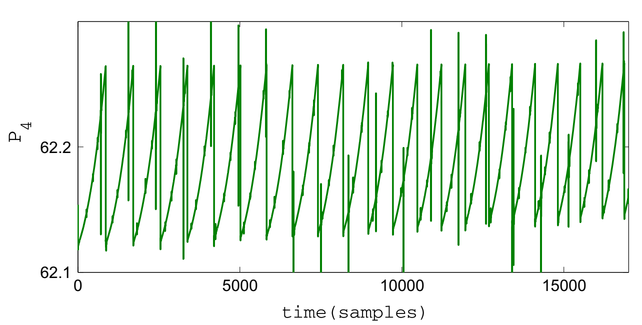

Fig. 2. Pressure measured in the membrane input.

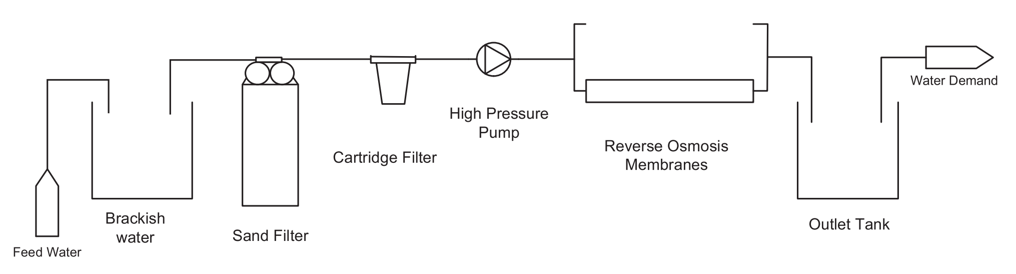

Fig. 1. Desalination plant scheme.

Table 1
Description of variables

	Name	Description	Units
1	P_3	Pressure in the cartridge filter input	bar
2	XS_1	Total solid concentration in the plant input	kg/m^3
3	XS_2	Total solid concentration in the cartridge filter input (sand filter output)	kg/m^3
4	P_1	Pressure in the plant input	bar
5	P_2	Pressure in the cartridge filter input (sand filter output)	bar
6	X_1	Salt concentration in the plant input	kg/m^3
7	P_4	Pressure in the membrane input	bar
8	Q_1	Flow in the plant input	m^3/d
9	X_2	Salt concentration in the membrane output	kg/m^3
10	Q_3	Flow in the membrane output	kg/m^3
11	Q_2	Flow in the plant output	m^3/d

measured variable set is formed by pressures, flows, total solid, and salt concentrations. Table 1 shows the name and the description of the variables.

Three types of fault were considered in the plant. One of them consists of an offset in the pressure sensor in the sand filter input (P_1). The other two faults are related with the membranes, concretely, faults simulated by altering several equations in the membrane model being a blockage and a breakage.

3. Principal component analysis

The PCA is a multivariate statistical technique. This technique is a versatile tool with several interesting properties. For example, (i) it is able to deal with data matrices with more variables than observations, (ii) it can handle missing data, and (iii) PCA can analyze highly correlated data and low rank data. PCA can reduce the dimensionality of the space of the measured variables from the process, normally highly correlated, transforming it into a new space of uncorrelated variables (components) which can be analyzed in order to understand and monitor the process or to detect faults and malfunctions in its behavior. All these properties are the reason why PCA and partial least squares regression are the most used multivariate statistical process control techniques.

Mathematically, PCA calculates the correlation structure of the process variables collected along the time. The process data are arranged in a matrix $\mathbf{X} \in \Re^{K \times J}$, where J is the number of process variables (variables) and K is the number of time samples (individuals). The data used to perform the PCA model for fault detection or monitoring scheme must be collected under normal plant operation.

PCA computes an approximation of the data matrix $\mathbf{X}$ as a product of two matrices $\mathbf{T}$ and $\mathbf{p}^T$. The columns $\mathbf{p}^T$ of the matrix $\mathbf{P}^T$ are known as the loading vectors; the columns ($\mathbf{t}$) of the matrix $\mathbf{T}$ are known as score vectors [24].

The matrix $\mathbf{X}$ is arranged with the data measured in the process. All these variables have different numerical ranges. The principal components analysis is based on the covariance matrix and is, therefore, variance dependent, and the variance is related with the numerical range. So, the columns (variables) of $\mathbf{X}$ has to be previously normalized to zero mean and unit variance.

The singular value decomposition can be calculated over the covariance matrix $\mathbf{S}$:

$$\mathbf{S} = \frac{1}{K-1}\mathbf{X}^T\mathbf{X} \tag{1}$$

obtaining [25]:

$$\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \tag{2}$$

where $\mathbf{\Lambda}$ is a diagonal matrix that contains the eigen values of $\mathbf{S}$ in its diagonal sorted in decreasing order $(\lambda_1, \lambda_2, \ldots, \lambda_{\text{rank}(\mathbf{X})})$.

The transformation, cited above, from the original space of the measured variables, normally highly correlated, to the uncorrelated reduced space of latent variables, is computed using the loadings matrix $\mathbf{P}_{1:A} \in \Re^{J \times A}$:

$$\mathbf{T} = \mathbf{X}\mathbf{P}_{1:A} \tag{3}$$

where the matrix $\mathbf{P}_{1:A}$ is formed by the A eigenvectors or columns of the matrix $\mathbf{V}$. These eigenvectors correspond to the greatest A eigenvalues λ_a, $a = 1, 2, \ldots, A$ in decreasing order in the diagonal of the matrix $\mathbf{\Lambda}$.

A reconstruction of the original variables can be calculated from the scores operating in Eq. 3:

$$\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}_{1:A}^T \tag{4}$$

The residual matrix $\mathbf{E}$ can be calculated as the difference between the original variables $\mathbf{X}$ and the variables estimated by the PCA model $\hat{\mathbf{X}}$:

$$\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}} \tag{5}$$

The complete PCA model can be expressed by means of two addends: the PCA model and the residuals:

$$\mathbf{X} = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a^T = \mathbf{T}\mathbf{P}_{1:A}^T + \mathbf{E} \tag{6}$$

The PCA model can be established using the nonlinear iterative partial least squares algorithm. This algorithm can deal with missing data [24].

There are several rules to choose the most suitable number of principal components A. The most used method to select the suitable number of principal components is the cross validation task [26,27]. Although, the best method to use is currently a matter of scientific debate [28,29].

3.1. Monitoring, FDI

Two control charts [6] are mainly used to monitor the state of the process. When the process data have been analyzed, the outliers are eliminated and the suitable number of principal components are computed, and new measured variables can be analyzed using the control charts based on the established PCA model. These control charts are drawn using two statistics: Hotelling's T^2 statistic and the square prediction error (Q) statistic, also called SPE. A fault detection and monitoring scheme can be performed using them in a real-time application connected to the plant.

Hotelling's T^2 statistic is based on the Mahalanobis distance and is computed as follows:

$$T^2 = \mathbf{x}^T \mathbf{P}_{1:A} \mathbf{\Lambda}_{1:A}^{-1} \mathbf{P}_{1:A}^T \mathbf{x} \tag{7}$$

where $\mathbf{\Lambda}_{1:A}$ is an $A \times A$ diagonal matrix of the higher A eigen values of the covariance matrix $\mathbf{S}$ in decreasing order along the diagonal.

The process state can be considered to be in normal operation conditions (NOC) state while the following inequation is satisfied:

$$T^2 \leqslant T_\alpha^2 = \frac{(K^2-1)A}{K(K-A)} F_\alpha(A, A-K) \tag{8}$$

where $F_\alpha(A, A-K)$ is the critical value $(100(1-\alpha))\%$ percentile) of the Fisher–Snedecor distribution or F-distribution with A and $K-A$ degrees of freedom and is the level of significance, usually taking values between 5 and 1%. This distribution is only valid for the T^2 statistic in case the tested sample is not used for calibration. If the sample is part of the calibration data set, the T^2 statistic is distributed as the beta-distribution [30].

Hotelling's statistic is based on the PCA model formed by the A principal components. The Q statistic is computed taking into account the rest of the components, and can therefore be used to detect deviations in the residuals:

$$Q = \mathbf{r}^T \mathbf{r} \tag{9}$$

with:

$$\mathbf{r} = (\mathbf{I} - \mathbf{P}_{1:A}\mathbf{P}\mathbf{T}_{1:A}^T)\mathbf{x} \tag{10}$$

The faulty states are detected in this statistic when the following in equation is not satisfied:

$$Q \leqslant Q_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{11}$$

with:

$$\theta_i = \sum_{j=A+1}^{J} \lambda_j^i \tag{12}$$

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \tag{13}$$

where c_α is the $100(1-\alpha)\%$ standardized normal percentile and α_j are the eigenvalues of the PCA residual covariance matrix $\mathbf{E}^T\mathbf{E}/(K-1)$.

The Q statistic should be checked first, if no point raises the control limit the process can be considered in-control. If the value of one of the statistics is greater than the upper limit, a diagnosis task must be performed to identify the variables related with the fault.

The PCA has been successfully applied to monitoring and fault detection tasks, but it does not provide quite enough information for fault isolation tasks. The contributions analysis [5,6] has been used as a first approximation to fault isolation. Contributions analysis calculates the influence of each one of the system's variables on the Q and T^2 statistics trigger.

4. Unfolded PCA

The plant studied in this work presents a behavior that can be approximated, from the signal point of view, to a batch process to batch processes. The database for performing the PCA model is made up of data from past NOC phases between cleaning cycles. For every $i = 1, 2, \ldots, I$ normal past batches $j = 1, 2, \ldots, J$ variables are collected at

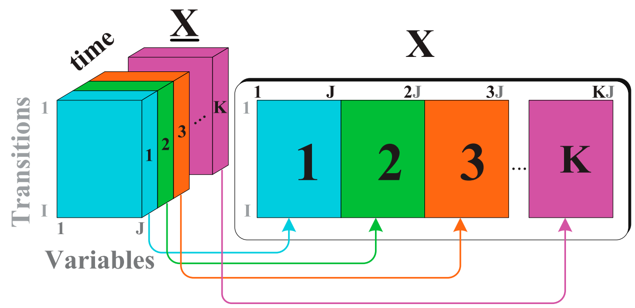

Fig. 3. Matrix unfolding.

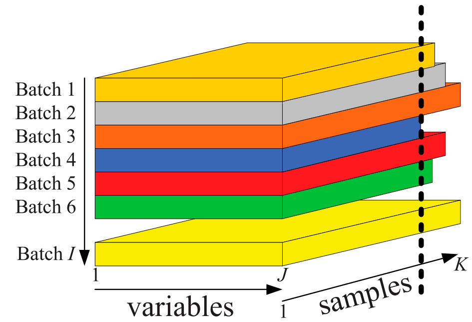

Fig. 4. Aligning batches with different length.

$k = 1, 2, \ldots, K$ time samples. All these data are ordered into a three-way matrix $\underline{\mathbf{X}}(I \times J \times K)$ (following the notation of [31]) as shown in the left part of the Fig. 3.

Several changes and variations are required to perform a FDI scheme in this type of processes based on UPCA due to fact that the database structure has changed. The principal modifications are resumed in the next subsections.

4.1. Unfolding the three-way data matrix

The unfolding procedure can be carried out in six particular ways [32]. Nomikos and MacGregor [11] and Kourti [20] recommend the batch-wise $(I \times KJ)$ option as the most appropriate and as a very good option for the online monitoring tasks. This is the option used in this work. It maintains the direction of the batches and all process variables at the first sample time are arranged into the unfolded matrix for all of the I NOC batches, this operation is repeated for every next sample time, as shown in Fig. 3.

The three-way data matrix has to be unfolded into a two-dimensional matrix to perform the PCA approach. This unfolding procedure can be performed in several ways: when the unfolding is carried out, one of the directions will remain unaltered, while the other direction will be the combination of the other two directions slice by slice.

When the matrix is unfolded the matrix normalization must be performed in the same way as the classical PCA, it means, to normalize every column to zero mean and unit variance.

4.2. Data alignment

When the data are collected from different batches, or cycles between cleaning phases, the number of samples of the data-sets can have different lengths. The reason for this is that the event that activates the cleaning phases is not a time event, it depends on other parameters, such as concentrations or pressures. When this phenomenon occurs, it is not possible to arrange the data matrix as shown in Fig. 4, due to the fact that the number of samples of the different data-sets is not the same.

The data can be truncated to the shortest length as a first solution, but this can produce loss of information, and the data matrix cannot reproduce the system state exactly. Two main techniques can be used to align the trajectories of the variables in order to arrange the three-way data matrix without loss of information. One of them is known as the indicator variable approach [11]. This scheme can be applied when a no noise, monotonically increasing or decreasing variable with the same starting and ending value can be found in the process; this variable, the indicator variable, leads the collection of the data instead of the time.

In this work, it is not so easy to find this indicator variable to perform the data alignment. When it is not possible to find an indicator variable, the dynamic time warping (DTW) approach [22] can be performed. For two multivariate trajectories of two different phases between cleaning cycles, $\mathbf{A} \in \Re^{K_1 \times J}$ and $\mathbf{B} \in \Re^{K_2 \times J}$, where the number of samples K_1 and K_2 are not equal, DTW method aligns these trajectories to the length of one of them or to a reference trajectory, by creating or eliminating the same points. This process of compression or expansion of the time scales must be performed by minimizing the dissimilarity between the two trajectories.

The iterative method for synchronizing batch trajectories was, cited previously, performed and applied. The DTW approach tries to find a path F with a specific number of points on a grid with dimensions $K_1 \times K_2$. The traced path on the grid must minimize the total distance between the trajectories. The DTW method provides a weight matrix which reveals the importance of the variables in the synchronization method. If the indicator variable approach is applied, the synchronization is led by a unique variable, while this procedure is distributed among several variables if the DTW approach is applied.

When a ruining cycle between two cleaning cycles is monitored online, the DTW approach finds the point in the reference trajectory with the least error with respect to the current time instant and performs the synchronization to this point. The future data of the running cycle are unknown, so it must be predicted, as the next section discusses. The online DTW was implemented following the guidelines of [22].

4.3. Imputation

During the online monitoring task, at each time sample t, the future value of the process variables is necessary to compute the scores and, therefore, the monitoring statistics. Fig. 5 shows this particular problem. The dimensions of the matrix $\mathbf{P}_{1:A}$ agree with the data matrix $\mathbf{X}$, which takes into account the process variables for all the time samples across past cycles under normal running. At the time instant t, shown in Fig. 5, only the sample times corresponding to the samples from 1 to t are available, and the future samples from $t+1$ to KJ are unknown. There are several methods to impute this future missing data. The imputation methods impute the future data and allow the scores to be estimated. The principal data imputation methods are presented in a clear and well-explained way in Ref. [33,34]. Two methods are considered in this work, the TRI and trimmed score regression method (TSR), because TRI is one of the most widely used methods and TSR presents good results without online considerations and an acceptable computational cost.

A specific nomenclature has to be cited after presenting the methods. As mentioned before, the data matrix $\mathbf{X}$ is arranged with column vectors $\mathbf{x}_j$ (variables) or row vectors $\mathbf{z}_i^T$ (observations). The loading matrix $\mathbf{P}$ are formed by the $\mathbf{p}_j$ as columns. The score matrix $\mathbf{T}$

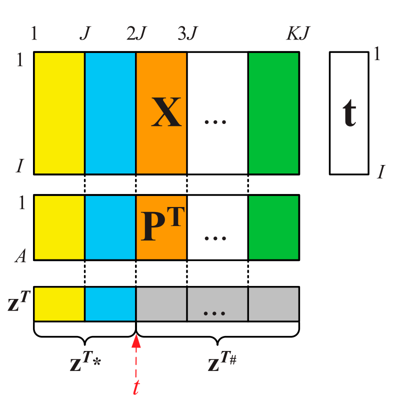

Fig. 5. Imputation.

can be considered as a set of row vectors $\boldsymbol{\tau}_i^T$ (scores in the ith observation) or column vectors $\mathbf{t}_i$ (latent variables or scores). A new observation at a particular time instant can be partitioned as follows:

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}^* \\ \mathbf{z}^\# \end{bmatrix} \tag{14}$$

where $\mathbf{z}^*$ is the current and past process variable (known) values and $\mathbf{z}^\#$ is the future data values (unknown).

The loadings matrix $\mathbf{P}$ can be partitioned, in the same way, by dividing the known and the unknown data and also dividing the A principal components and leaving $K-H$ components, where $H = \text{rank}(\mathbf{X})$:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^* \\ \mathbf{P}^\# \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{1:A} & \mathbf{P}_{A+1:H} \end{bmatrix} = \begin{bmatrix} \mathbf{P}^*_{1:A} & \mathbf{P}^*_{A+1:H} \\ \mathbf{P}^\#_{1:A} & \mathbf{P}^\#_{A+1:H} \end{bmatrix} \tag{15}$$

The first cited is the TRI. This substitutes the missing values by its mean, and since the monitoring statistics are computed using centred data, the mean value is zero. This means that this method substitutes the unknown future data by zeros $\mathbf{z}^\# = 0$. And the scores can be estimated as [33]:

$$\begin{aligned} \hat{\boldsymbol{\tau}}_{1:A} &= \mathbf{P}_{1:A}^T\mathbf{z} = \begin{bmatrix} \mathbf{P}^{*T}_{1:A} & \mathbf{P}^{\#T}_{1:A} \end{bmatrix} \begin{bmatrix} \mathbf{z}^* \\ \mathbf{z}^\# \end{bmatrix} \\ &= \mathbf{P}^{*T}_{1:A}\mathbf{z}^* + \mathbf{P}^{\#T}_{1:A}\mathbf{z}^\# = \mathbf{P}^{*T}_{1:A}\mathbf{z}^* \end{aligned} \tag{16}$$

The covariance matrix for the score vector estimation is calculated as:

$$\begin{aligned} \text{Var}(\boldsymbol{\tau}_{1:A} - \hat{\boldsymbol{\tau}}_{1:A}) = {} & \mathbf{P}^{*T}_{1:A}\mathbf{P}^*\boldsymbol{\Lambda}\,\mathbf{P}^{*T}\mathbf{P}^*_{1:A} + \boldsymbol{\Lambda}_{1:A} \\ & - \mathbf{P}^{*T}_{1:A}\mathbf{P}^*_{1:A}\boldsymbol{\Lambda}_{1:A} - \boldsymbol{\Lambda}_{1:A}\mathbf{P}^{*T}_{1:A}\mathbf{P}^*_{1:A} \end{aligned} \tag{17}$$

The other method considered is the TSR. In this case, the method reconstructs the scores $\mathbf{T}_{1:A}$ from the trimmed scores ($\mathbf{T}^*_{1:A}$) using the following regression structure $\mathbf{T}_{1:A} = \mathbf{T}^*_{1:A}\,\mathbf{B} + \mathbf{U}$. The scores are estimated as follows [33]

$$\hat{\boldsymbol{\tau}}_{1:A} = \boldsymbol{\Lambda}_{1:A}\mathbf{P}^{*T}_{1:A}\mathbf{P}^*_{1:A}\left(\mathbf{P}^{*T}_{1:A}\mathbf{P}^*\boldsymbol{\Lambda}\mathbf{P}^{*T}\mathbf{P}^*_{1:A}\right)^{-1}\mathbf{P}^{*T}_{1:A}\mathbf{z}^* \tag{18}$$

For the TSR imputation method, the covariance matrix for the score vector estimation error is calculated as:

$$\begin{aligned} & \text{Var}(\boldsymbol{\tau}_{1:A} - \hat{\boldsymbol{\tau}}_{1:A}) \\ & = \left[\mathbf{I}_A - \boldsymbol{\Lambda}_{1:A}\mathbf{P}^{*T}_{1:A}\mathbf{P}^*_{1:A}\left(\mathbf{P}^{*T}_{1:A}\mathbf{P}^*\boldsymbol{\Lambda}P^{*T}\mathbf{P}^*_{1:A}\right)^{-1}\mathbf{P}^{*T}_{1:A}\mathbf{P}^*_{1:A}\right]\boldsymbol{\Lambda}_{1:A} \end{aligned} \tag{19}$$

4.4. Control limits

The T^2 statistic and the upper limit, necessary for building the monitoring control charts, can be computed using Eqs. (7) and (8). In this particular case, the number of individuals in the unfolded matrix $\mathbf{X}$ is the number of past cycles considered I, instead of the number of process samples K.

The Q statistic for this approach, as Nomikos and MacGregor [11] propose, can be substituted by the squared prediction error Q in every sample time k (Q_k):

$$Q_k = \sum^{J} \mathbf{e}(c)^2 \tag{20}$$

instead of the squared residuals over all time periods Q, as this measurement does not represent the instantaneous perpendicular distance to the reduced space, where e is calculated for the current measures vector at instant k $\mathbf{z}^T_{new,t}(1 \times J)$ (scaled) as:

$$\mathbf{e} = \mathbf{z}^T_{new,t} - \hat{\mathbf{z}}^T((k-1)J + 1 : kJ) \tag{21}$$

with

$$\hat{\mathbf{z}} = \hat{\boldsymbol{\tau}}_{1:A}\mathbf{P}^T_{1:A}$$

The control limit for this statistic can be computed by approximating the value in every sample to a chi-squared distribution, calculated for a specific level of significance as:

$$Q_{\alpha k} = g_k \chi^2_{h_k,\alpha} \tag{22}$$

where $\chi^2_{h_k,\alpha}$ is the limit value of the chi squared variable with h_k degrees of freedom and level of significance. In this equation, g_k and h_k can be approximated in several ways, as Nomikos and MacGregor [11] discuss and Lennox et al. [35] compare. In this work, these parameters are approximated using the mean (m_k) and the variance (v_k) of the $k-$th sample of the reference nominal data-sets used to compute the UPCA model. The approximation can be computed using the following expressions:

$$g_k = \frac{v_k}{2m_k} \tag{23}$$

$$h_k = \frac{2m_k^2}{v_k} \tag{24}$$

The parameters (m_k) and (v_k) for the control limit at each time instant k can be calculated using the immediately previous and later time observations $k-1, k-2, k, k+1, k+2$, that is, by using a moving window for the estimations of the control limit for the instant in the center of this window.

4.5. Contribution plots

The contribution plot of the normalized error in this case is the same bar plot as in the case of the classical PCA approach. If the Q statistic rises above the established upper limit, the normalized errors at that time instant k are shown in a bar plot.

However, the contributions plot of the normalized scores in this case is not a very helpful tool for fault diagnosis tasks. When the imputation procedure is run, the value of the scores corresponds to the scores of the whole phase between cleaning cycles. These greatest scores observed in this plot could be used to compute the variable contribution to these scores. In this case, a vector of variable contributions with $kJ \times 1$ dimensions is obtained. A useful plot can be the variable contributions to the scores with the greatest values from the beginning of the running cycle to the current time instant ($kJ \times 1$) grouped by each process variable in a different plot. In each of these plots, the variable contributions are arranged in time order, and the operator can see the evolution of the variable contribution and detects which of them are involved with the greatest scores that have caused an abnormal value in the T^2 statistic. An example of this plot can be seen in Fig. 15.

5. Results and discussion

This system does not run in a constant steady due to the different particularities cited. So, the classical PCA scheme for monitoring and fault detection, described in section 3 and applied to this type of systems, is not the most suitable solution. If the plant is monitored using the T^2 statistic (Fig. 6(a)) and the Q statistic (Fig. 6(b)) with a classical PCA approach, a high number of false alarms appear, as both graphics show. The PCA model built to perform this monitoring task was arranged with nominal data from eleven variables measured during several running cycles and different cleaning cycles. The monitoring scheme was applied to new data collected from the simulated plant.

If the UPCA approach is applied to this case study, first, a rich database has to be arranged. This database is formed by the variables measured in dif-

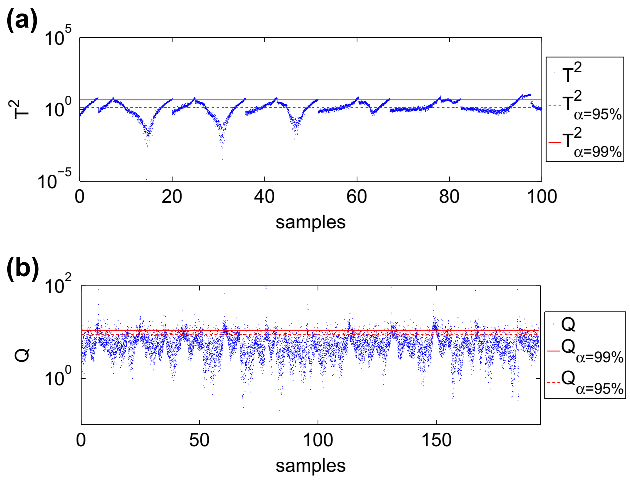

Fig. 6. Monitoring using the classical PCA approach (a) T^2 monitoring statistic and (b) Q monitoring statistic.

ferent past implementations between the cleaning cycles under NOCs.

The values of the variables along time could be ordered into the same matrix in order to apply the UPCA approach as a first approximation for each past running cycle considered, following the structure shown in Fig. 3. However, the cleaning phase frequency is not the same, due to the different nature of the filters and the membrane. This phenomenon represents a new problem because, in the same way as applying classical PCA, these changes in the behavior produced by the different cleaning cycles can be detected by the monitoring statistics as faults. So, there is no single criterion for ordering the data into the three-way matrix because of the different cleaning cycle frequencies.

The solution proposed to deal with this drawback consists of arranging three different UPCA models. One specific model for variables related with the membrane and another two models for the two filters, respectively.

Table 2 shows the main characteristics of the three UPCA models considered in this work. The second column of this table shows the variables related with this specific section. The variables that are not affected by any of the cleaning cycles can be considered in several models, if they are related with two or more subsystems. Therefore, the variables affected by a particular cleaning cycle running could only be included in that specific model. The third column shows the number of principal components considered using a cross-validation procedure. The last column shows the interval of the length of the different cycles between cleaning cycles for every subsystem. This variable organization can considerably reduce the number of false alarms and it allows the three critical parts of the plant to be monitored separately. Also, this configuration can be seen as a step towards distributed fault detection.

Table 2
UPCA models

UPCA model	Variables name	Number of components	Cycles length
Membranes	P_2, X_1, P_4, Q_1, X_2, Q_2 and Q_3	10	840–860
Sand filter	X_{S1}, X_{S2}, P_1,P_2, X_1 and Q_1	8	590–610
Cartridge filter	P_3, P_2, X_1 and Q_1	13	675–858

The cleaning cycles are not performed with a determined time frequency. The running of these cycles depends on several variables like pressures or concentrations. When one or more variables rise above a determined threshold, an event is trigged and the cleaning phase begins. This configuration explains why the running cycles considered to arrange the three-way matrix do have not the same number of samples, and an alignment method has to be performed to align all these cycles and to build the UPCA model. As cited in the previous section, the DTW approach was applied to obtain this objective.

Fig. 7 shows the nonsteady behavior of some of the variables during different running cycles. The nonequal duration of the different running cycles can also be seen in these graphics. The variables, after being synchronized to perform the three-way matrix needed for applying UPCA, are shown in Fig. 8. This graphic shows that the signals are slightly deformed during the alienation procedure. The variables were synchronized to the mean length, and the added points were principally added at the beginning of the signals.

Fig. 9 shows the evolution of the weights of the different variables along the iterations of the DTW algorithm until convergence. The variable with the greatest importance to lead the synchronization is clearly the concentration X_2. Using an indicator variable approach, this variable would lead all the alienation, but in this case, the pressure P_4 is not so important, but it still has considerable weight. The flows Q_3 and Q_2 are also considerable weight, but little importance. The rest of the variables are not taken into account to lead the alienation by the algorithm.

A suitable parameter to compare the imputation error, as Arteaga and Ferrer [34] suggest, can be the prediction error sum of variances (PRESV). It is com-

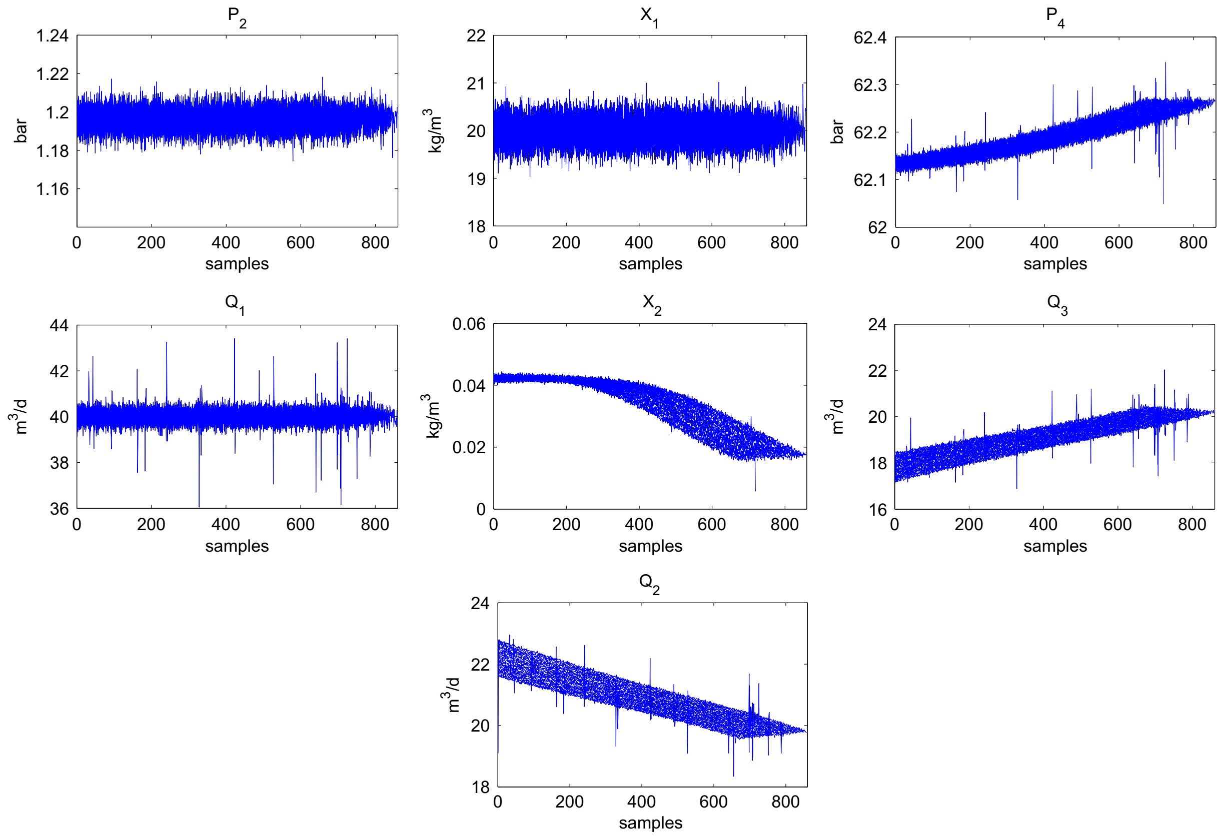

Fig. 7. Measured variables related with the membrane.

puted as the trace of the covariance matrix for the score vector estimation error (Eqs. (17) and (18)). During the plant monitoring, the future samples $(K-k)$ at sample time k $(k<K)$ are not known. Fig. 10 shows the PRESV values at different sample times for both imputation methods TRI and TSR for the monitoring of the membrane section. Nine different time points along a cycle between two cleaning phases are considered ($k_1 = 599,\ k_2 = 1,198,\ \ldots,\ k_9 = 5,391$), using the data-set of past NOC cycles. The estimation error decays along the cycle due to the decrease in the unknown data as Fig. 10 shows.

Fig. 10 shows that the TSR method presents better results than the TRI method, despite the fact that the TRI method is usually more widely used than the TSR method. The TSR method presents less PRESV without a significant computational cost increment. Similar results were obtained for the sand filter and the carriage filter. So, in this article, the TSR method is the method selected to impute the future unknown data.

The reduction in the false alarms ratio using the explained approach is summarized in Table 3. This table shows the false alarms ratio obtained using the UPCA approach to the three considered sections and applying the classical PCA to all variables and to the three sections. The UPCA method obtains a decrease in the number of false alarms in the T^2 monitoring statistic. A reduction in the Q statistic is also observed. The decrease is principally obtained in the membrane section.

A monitoring using T^2 of the membrane section applying UPCA is shown in Fig. 11(a). False alarms principally appear in the first samples, and in the rest of the monitoring performance, the statistic remains under the control limits. This can be due to the $\mathbf{P}^{*T}_{1:A}\mathbf{P}^{*}\mathbf{\Lambda}P^{*T}\mathbf{P}^{*}_{1:A}$ matrix in Eq. (18), which may be ill conditioned at the beginning of the cycle because the known data are scant.

Fig. 11(b) shows the control plot using the Q monitoring statistic. The upper limits, with $\alpha = 99\%$, is equalled to one and the other control limit and the monitoring statistic are divided by the $\alpha = 99\%$ upper limit to normalize this control plot, because the control limits in this statistic are not constant and the plot cannot be user-friendly.

Table 4 shows the main results achieved in the fault detection task. For each considered fault, four

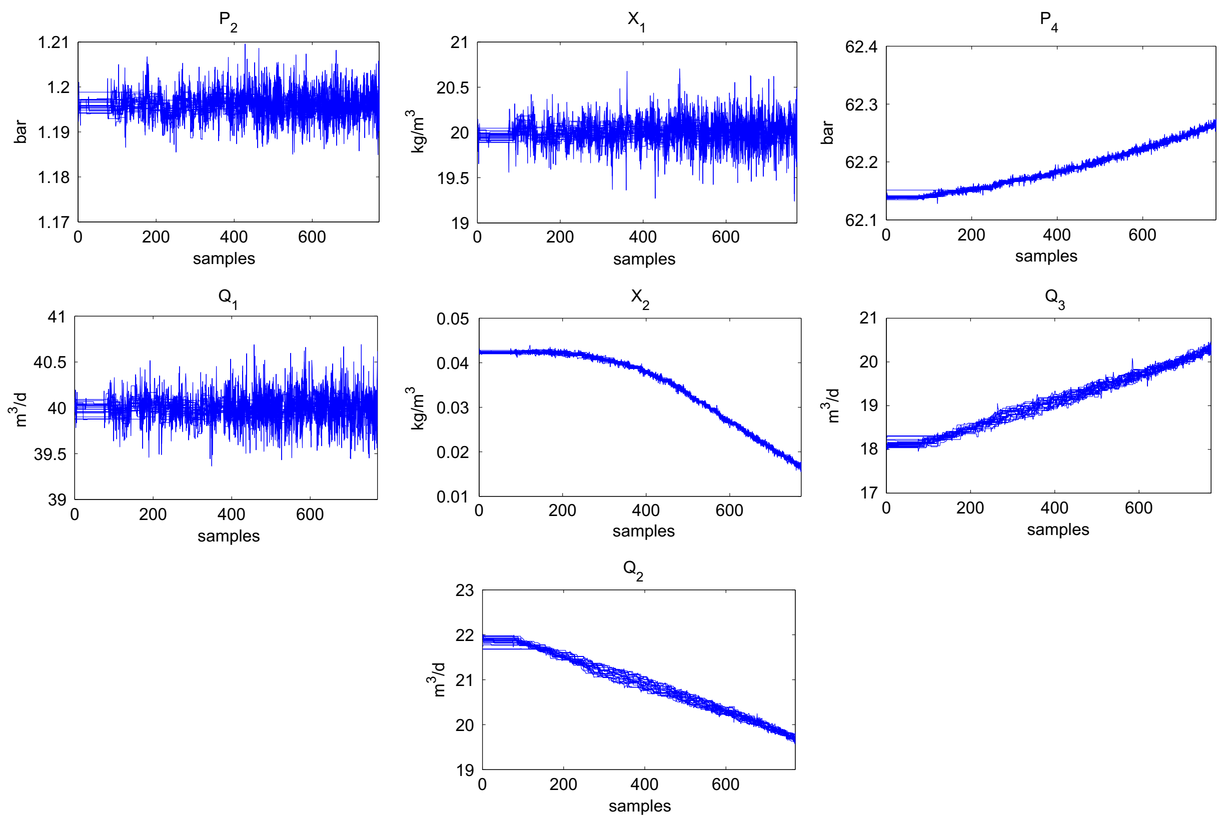

Fig. 8. Membrane variables synchronized.

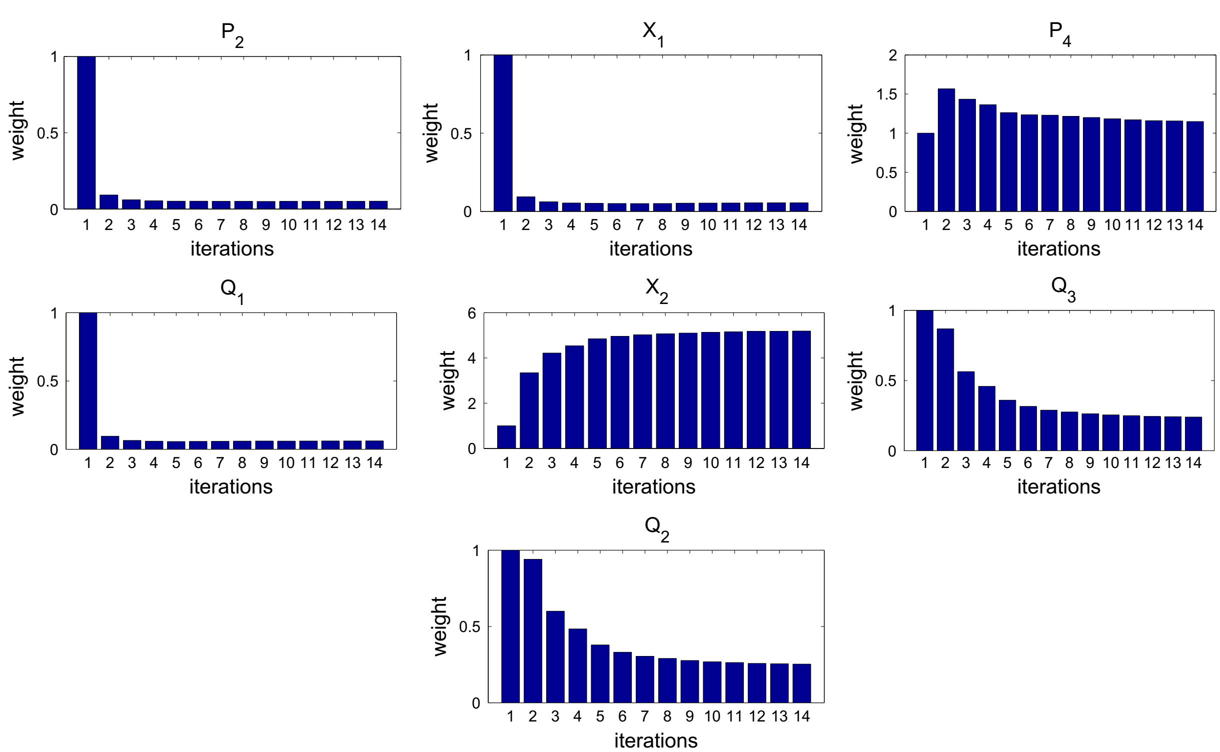

Fig. 9. DTW variable weights.

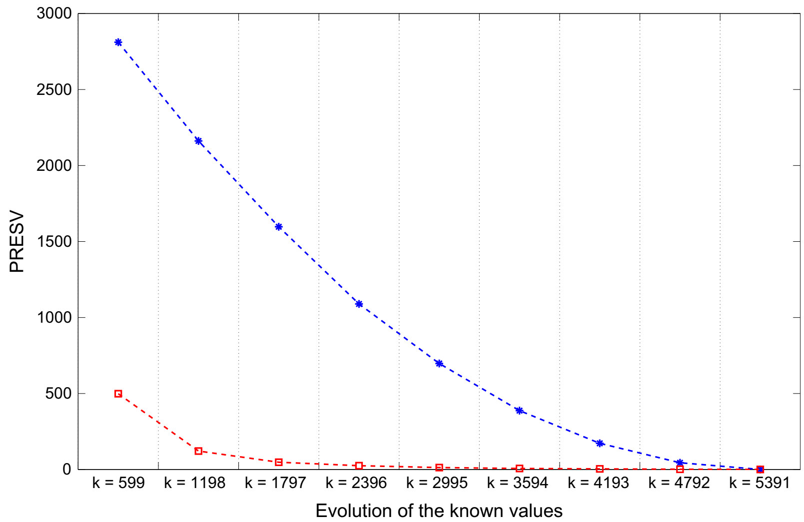

Fig. 10. PRESV for both imputation methods considered at different sample points k. TSR method results are plotted with squares and a dashed line and TRI method results are plotted with asterisks and a dashed line.

Table 3
False alarms percentage

Method	Percentage T^2 (%)	Q (%)
Classical PCA		
All variables	6.7	6.7
Sand filter	10.5	11.6
Cartridge filter	8.8	12.9
Membrane	9.4	10.5
UPCA (TSR)		
Sand filter	1.6	10.2
Cartridge filter	2.6	11.3
Membrane	1.0	4.0

Table 4
Detection results

Fault	Size (%)	Statistic	Delay
Membrane breakage	10	Q	Instantaneous
	20	Q	Instantaneous
	40	Q	Instantaneous
	60	Q	Instantaneous
Membrane blockage	10	Q	Instantaneous
	20	Q	Instantaneous
	40	Q	Instantaneous
	60	Q	Instantaneous
Sensor offset	10	Q	Instantaneous
	20	Q	Instantaneous
	40	Q	Instantaneous
	60	Q	Instantaneous

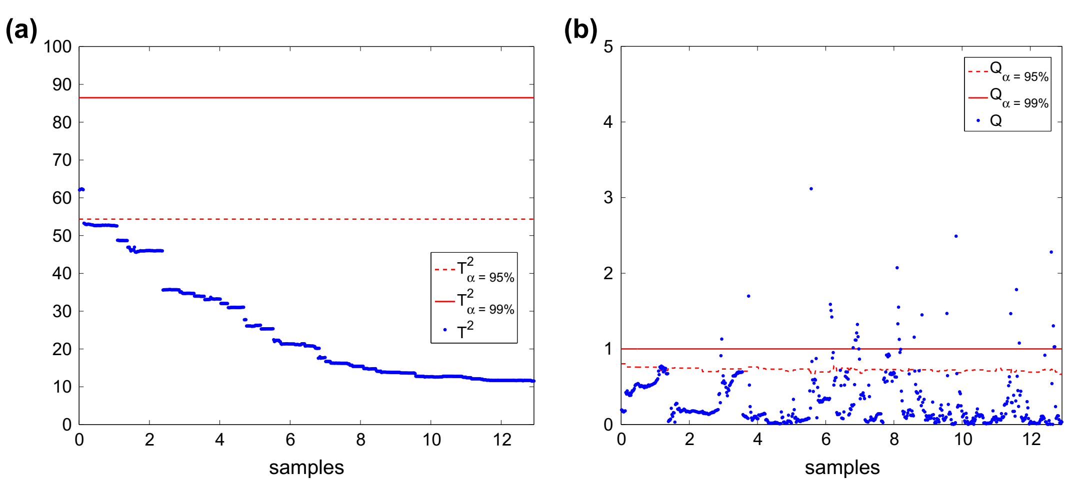

Fig. 11. Monitoring using UPCA in the membrane section (a) T^2 monitoring statistic and (b) Q monitoring statistic.

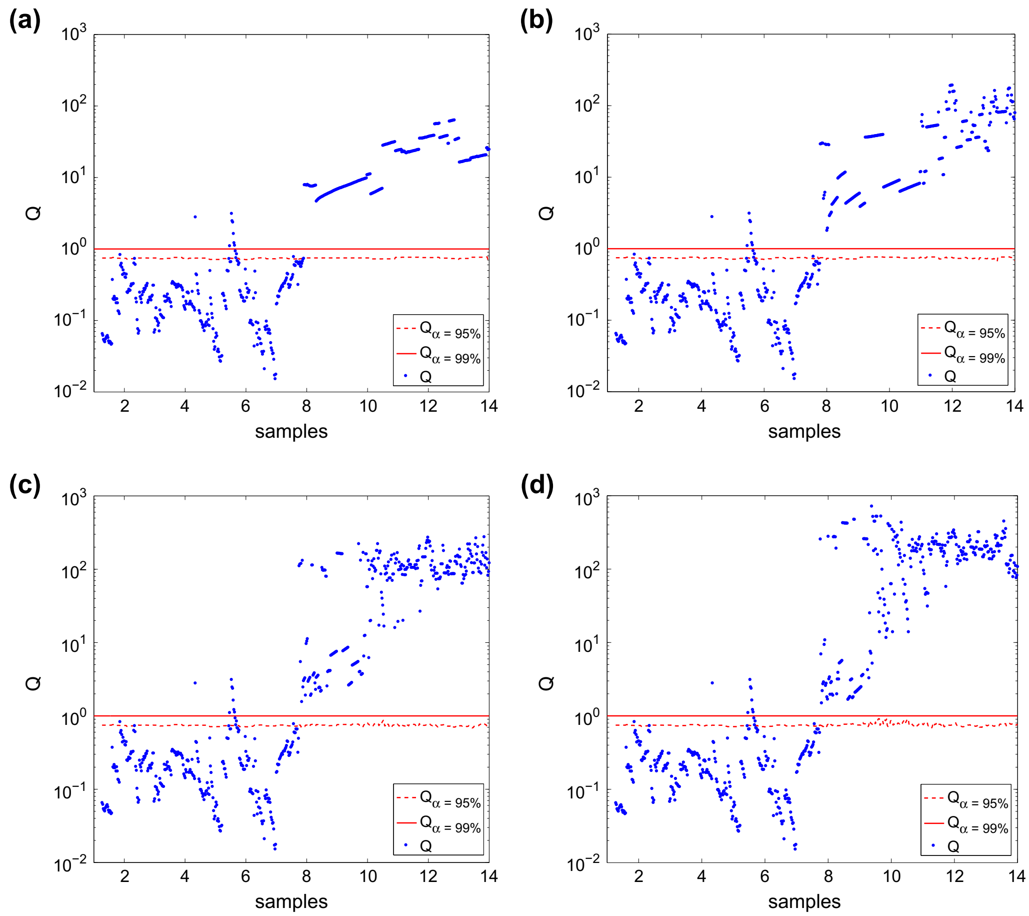

Fig. 12. Fault detection of a breakage in the membrane with different fault sizes: (a) 10%, (b) 20%, (c) 40% and (d) 60%.

size faults were considered. The third column (statistics) shows what statistic first detected the fault. The *Q* statistic detected the faults first in all cases. The fourth column shows the time delay in fault detection. In this work, all fault considered are abrupt faults, which is why there is no delay in the fault detection. Fig. 12(a–d) show the monitoring of a breakage in the membrane with different fault sizes.

When a fault is detected, contributions can be plotted to identify which variables are involved with the fault. This is not a complete diagnosis procedure, but it can be very useful to perform the diagnosis and isolation tasks by the plant operators. Fig. 13 shows the contribution plot of the normalized errors of the variables plotted after the detection of a breakage (20%) in the membrane. The figure shows that the principal variables related with this fault are the flows measured after and before the membrane. One of the pressures and one of the concentrations present anomalous values too.

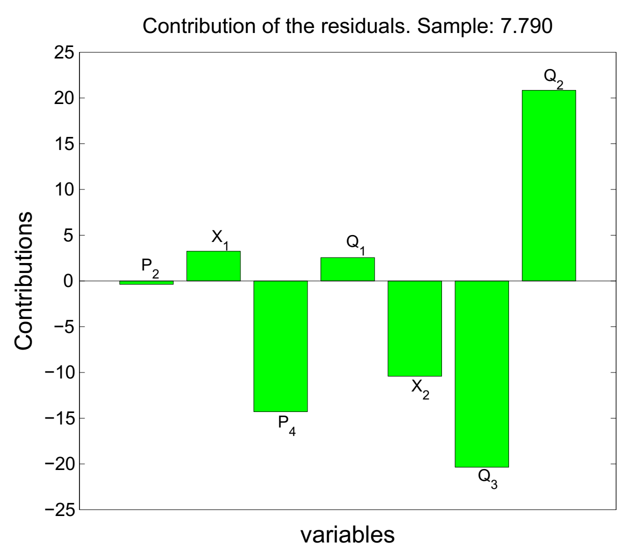

Fig. 13. Contribution plot of the normalized errors of the variables of a breakage in the membrane.

The faults are detected by the Q statistic in all the cases and it is not necessary to inspect the T^2, and therefore the contribution analysis related with this statistic is not necessary. Despite this, Figs. 14 and 15 show an example of the contribution plot of normalized scores and the variable contributions to the three greatest scores plot; in the last case, the contributions are grouped by variables in order to show the evolution of the variables along the cycle and identify what variables present an abnormal value.

6. Conclusions

PCA has been widely used as monitoring and fault detection technique. It is not complex to implement fault detection techniques based on PCA and it presents good results in many processes.

The PCA-based method shows the best results when the process to monitor is a continuous process that presents a constant steady state. Process with multiple operating points or processes whose state can be changed during the operation do not present good results due to these changes can be detected as faults by the PCA-based approaches.

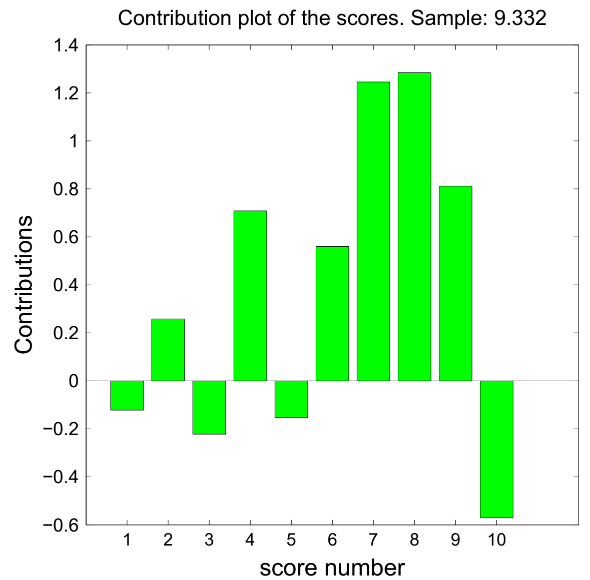

Fig. 14. Contribution bar plot of normalized scores corresponding to a breakage in the membrane.

In this case, the considered RO plant does not present a continuous operating point due to the plant does

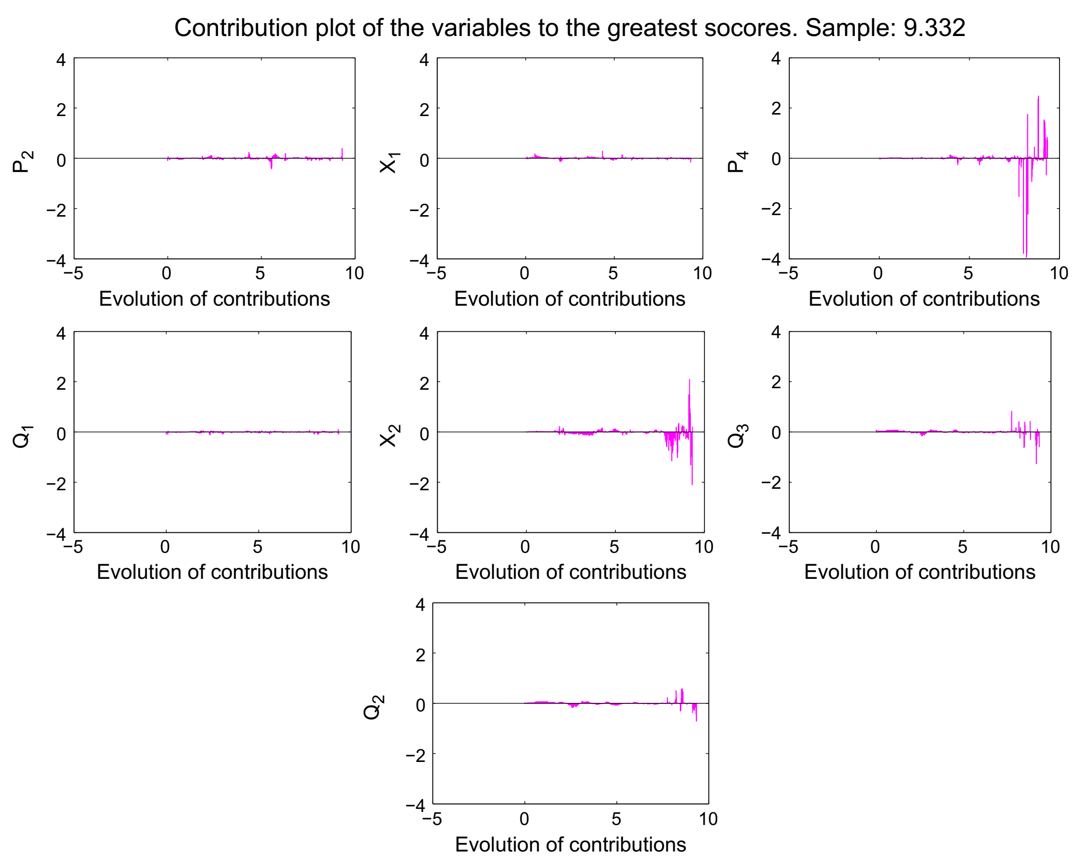

Fig. 15. Variable contributions bar plot to the three greatest scores corresponding to a breakage in the membrane.

not present the same behavior from the point of view of the data-driven model that PCA builds. This behavior is not constant due to the required cleaning phases. So, the use of UPCA, a method defined for batch processes is proposed in this work to monitor this type of plants. As this process is a continuous one, some modifications in the UPCA method have been taken into account, particularly the alignment task, different imputation method, and different contribution plots.

The main result is a reduction in the false alarms ratio. The faults considered are detected and the possibility of performing the contribution analysis is also presented. So the main conclusion is that UPCA based techniques can be a good solution for monitoring and detecting faults in this type of plants.

Acknowledgments

This work has been supported in part by the national research agency of Spain (CICYT) through the project DPI2009-14410-C02-02.

References

[1] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, Diagnosis and Fault Tolerant Control, Springer, Berlin, 2003.
[2] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S.N. Kavuri, A review of process fault detection and diagnosis. Part I: Quantitative model-based methods, Comput. Chem. Eng. 27 (3) (2003) 291–311.
[3] V. Venkatasubramanian, R. Rengaswamy, N. Surya, A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies, Comput. Chem. Eng. 27(3) (2003) 313–326.
[4] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, K. Yin, A review of process fault detection and diagnosis. Part III: Process history based methods, Comput. Chem. Eng. 27(3) (2003) 327–346.
[5] T. Kourti, J.F. MacGregor, Multivariate SPC methods for process and product monitoring, J. Qual. Technol. 28(4) (1996) 409–428.
[6] A. Ferrer, Multivariate statistical process control based on principal component analysis (MSPC-PCA): Some reflections and a case study in an autobody assembly process, Qual. Eng. 19(4) (2007) 311–325.
[7] D.X. Tien, K.W. Lim, L. Jun, Compartive study of PCA approaches in process monitoring and fault detection, in: Proceedings of the 30th Annual Conference of IEEE Industrial Electronics Society, Busan, South Korea, 2004.
[8] W. Li, H.H. Yue, S. Valle-Cervantes, S.J. Qin, Recursive PCA for adaptive process monitoring, J. Process Control 10 (2000) 471–486.
[9] S. Wold, Exponentially weighted moving principal components analysis and projections to latent structures, Chemom. Intell. Lab. Syst. 23(1) (1994) 149–161.
[10] M.A. Kramer, Nonlinear principal component analysis using autoassociative neural network, AIChE J. 37(2) (1991) 233–243.
[11] P. Nomikos, J.F. MacGregor, Multivariable SPC charts for monitoring batch processes, Technometrics 37(1) (1995) 41–59.
[12] P. Nomikos, Detection and diagnosis of abnormal batch operations based on multi-way principal component analysis, ISA Trans. 35(3) (1996) 259–266.
[13] D. Aguado, A. Ferrer, J. Ferrer, A. Seco, Multivariate SPC of a sequencing batch reactor for wastewater treatment, Chemom. Intell. Lab. Syst. 85(1) (2007) 82–93.
[14] K. Villez, K. Steppe, Dirk J.W. De Pauw, Use of unfold PCA for online plant stress monitoring and sensor failure detection, Biosystems Eng. 103(1) (2009) 23–34.
[15] C.W. McFalla, A. Bartman, P.D. Christofides, Y. Cohen, Control and monitoring of a high recovery reverse osmosis desalination process, Ind. Eng. Chem. Res. 47(17) (2008) 6698–6710.
[16] A. Sallami, A. Ben Chaabene, A. Sallami, Robust fault diagnosis of a reverse osmosis desalination system modeled by bond graph approach, Int. J. Comput. Sci. Netw. Secur. 11(12) (2011) 105–111.
[17] A. Gambier, T. Miksch, E. Badreddin, Fault-tolerant control of a small reverse osmosis desalination plant with feed water bypass, in: American Control Conference, Baltimore, MD, 2010.
[18] A. Gambier, N. Blümlein, E. Badreddin, Real-time fault tolerant control of a reverse osmosis desalination plant based on a hybrid system approach, in: American Control Conference, St. Louis, MO, 2009.
[19] K. Bourouni, Availability assessment of a reverse osmosis plant: Comparison between reliability block diagram and fault tree analysis methods, Desalination 313 (2013) 66–76.
[20] T. Kourti, Multivariable dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions, J. Chemom. 17(1) (2003) 93–109.
[21] Y. Zhang, M.S. Dudzic, Online monitoring of steel casting processes using multivariate statistical technologies: From continuous to transitional operations, J. Process Control 16(8) (2006) 819–829.
[22] A. Kassidas, J.F. MacGregor, P.A. Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE J. 44(4) (1998) 864–875.
[23] L.G. Palacín, F. Tadeo, J. Salazar, C. de Prada, Operation of desalination plants using renewable energies and hybrid control, Desalin. Water Treat. 25(1–3) (2011) 119–126.
[24] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemom. Intell. Lab. Syst. 2(1–3) (1987) 37–52.
[25] L.H. Chiang, E.L. Russell, R.D. Braatz, Fault Detection and Diagnosis in Industrial Systems, Springer, New York, NY, 2001.
[26] H.T. Eastment, W.J. Krzanowski, Cross-validatory choice of the number of components from a principal component analysis, Technometrics 24(1) (1982) 73–77.
[27] R. Bro, K. Kjeldahl, A.K. Smilde, H.A. Kiers, Cross-validation of component models: A critical look at current methods, Anal. Bioanal. Chem. 390(5) (2008) 1241–1251.
[28] J. Camacho, New methods based on the projection to latent structures for monitoring, prediction and optimization of batch processes, PhD thesis, Universidad Politécnica de Valencia, Valencia, Spain, 2007.
[29] J. Camacho, J. Picó, A. Ferrer, Cross-validation in principal component analysis: Searching for the 'best' approach, in: Chemometrics in Analytical Chemistry, Antwerp, 2010.
[30] N.D. Tracy, J.C. Young, R.L. Mason, Multivariate control charts for individual observations, J. Qual. Technol. 24 (1992) 88–95.
[31] H.A.L. Kiers, Towards a standardized notation and terminology in multiway analysis, J. Chemom. 14(3) (2000) 105–122.
[32] M. Zarzo, A. Ferrer, Batch process diagnosis: PLS with variable selection versus block-wise PCR, Chemom. Intell. Lab. Syst. 73(1) (2004) 15–27.
[33] F. Arteaga, A. Ferrer, Dealing with missing data in MSPC: Several methods, different interpretations, some examples, J. Chemom. 16 (2002) 408–418.
[34] F. Arteaga, A. Ferrer, Framework for regression-based missing data imputation methods in on-line MSPC, J. Chemom. 19 (2005) 439–447.
[35] B. Lennox, G.A. Montague, H.G. Hiden, G. Kornfeld, P.R. Goulding, Process monitoring of an industrial fed-batch fermentation, Biotechnol. Bioeng. 74(2) (2001) 125–135.